mri_dat_analysis_four

Philip Anderson; panders2@tamu.edu

6/30/2018

Objective

This analysis will synthesize a lot of the work that has been done so far

a) Data Prep + EDA

b) Model Fitting

b2) exogenous predictors

c) Forecasting

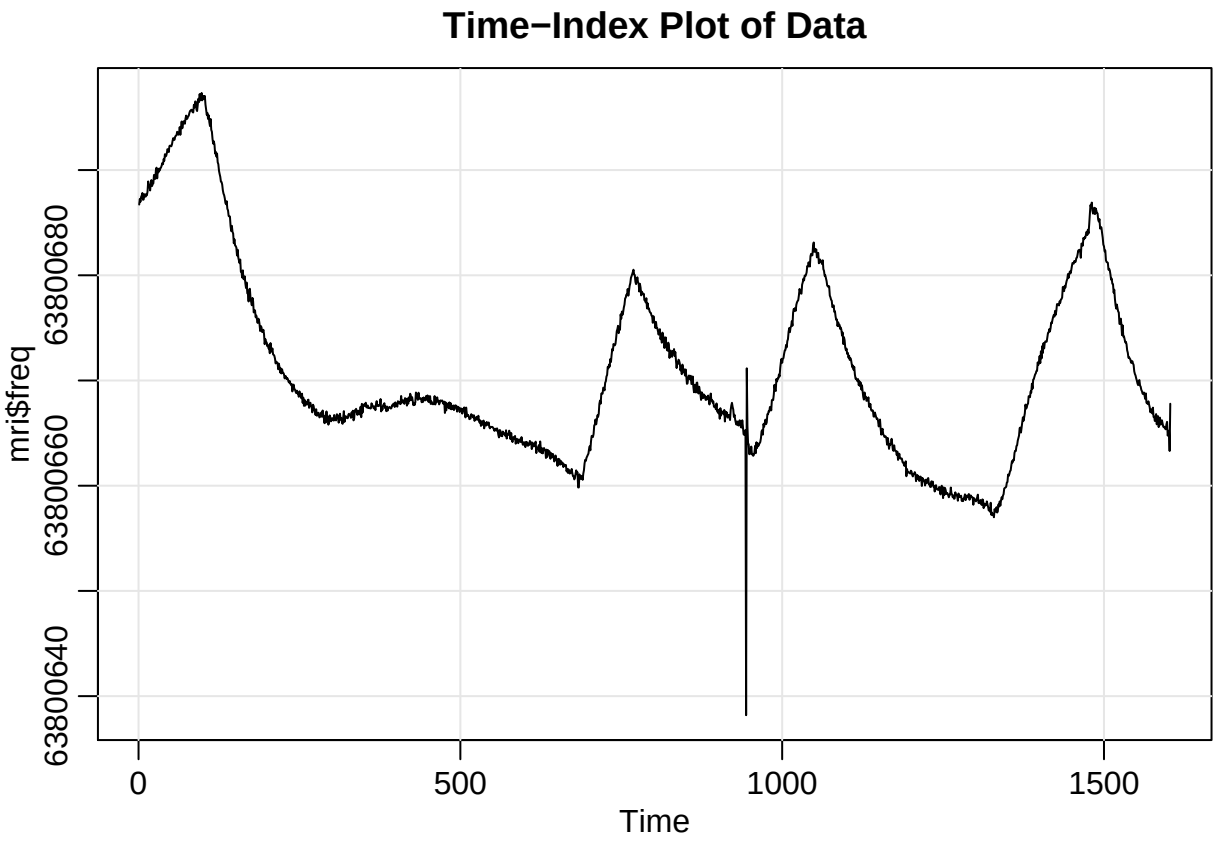
d) Seasonal Decomposition

Data Prep + EDA

```
#read in the data
mri <- read.csv("/Users/panders2/Documents/schools/tamu/stat_626/project/stat_626_proj/mri_dat_one.csv")
names(mri) <- c("hour", "minute", "freq", "int_pressure", "atm_pressure", "tot_pressure", "tesla")
str(mri)
```

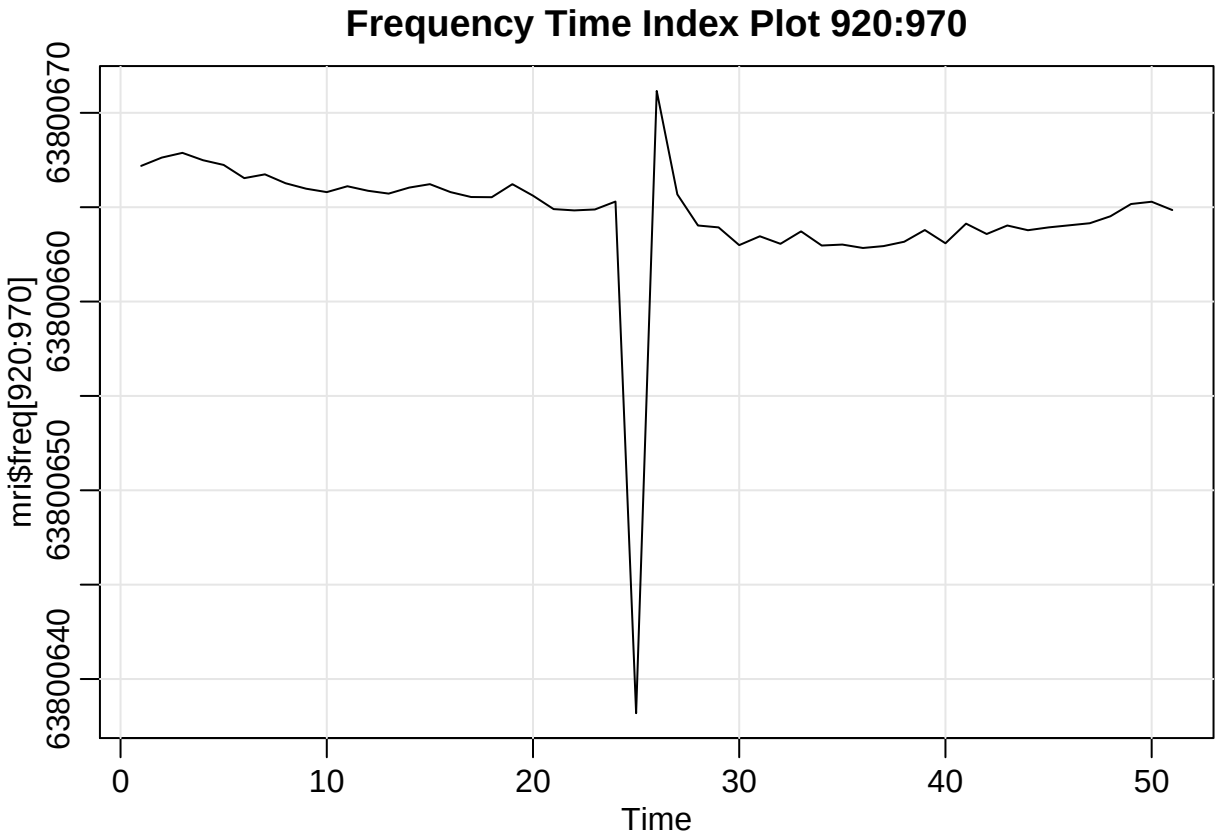
```
## 'data.frame':    1603 obs. of  7 variables:
##  $ hour        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ minute      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ freq        : num  63800687 63800687 63800687 63800687 63800688 ...
##  $ int_pressure: num  2.95 2.95 2.96 2.97 2.97 ...
##  $ atm_pressure: num  14.7 14.7 14.7 14.7 14.7 ...
##  $ tot_pressure: num  17.6 17.6 17.6 17.6 17.7 ...
##  $ tesla       : num  1.5 1.5 1.5 1.5 1.5 ...
```

```
astsa::tsplot(mri$freq, main="Time-Index Plot of Data")
```



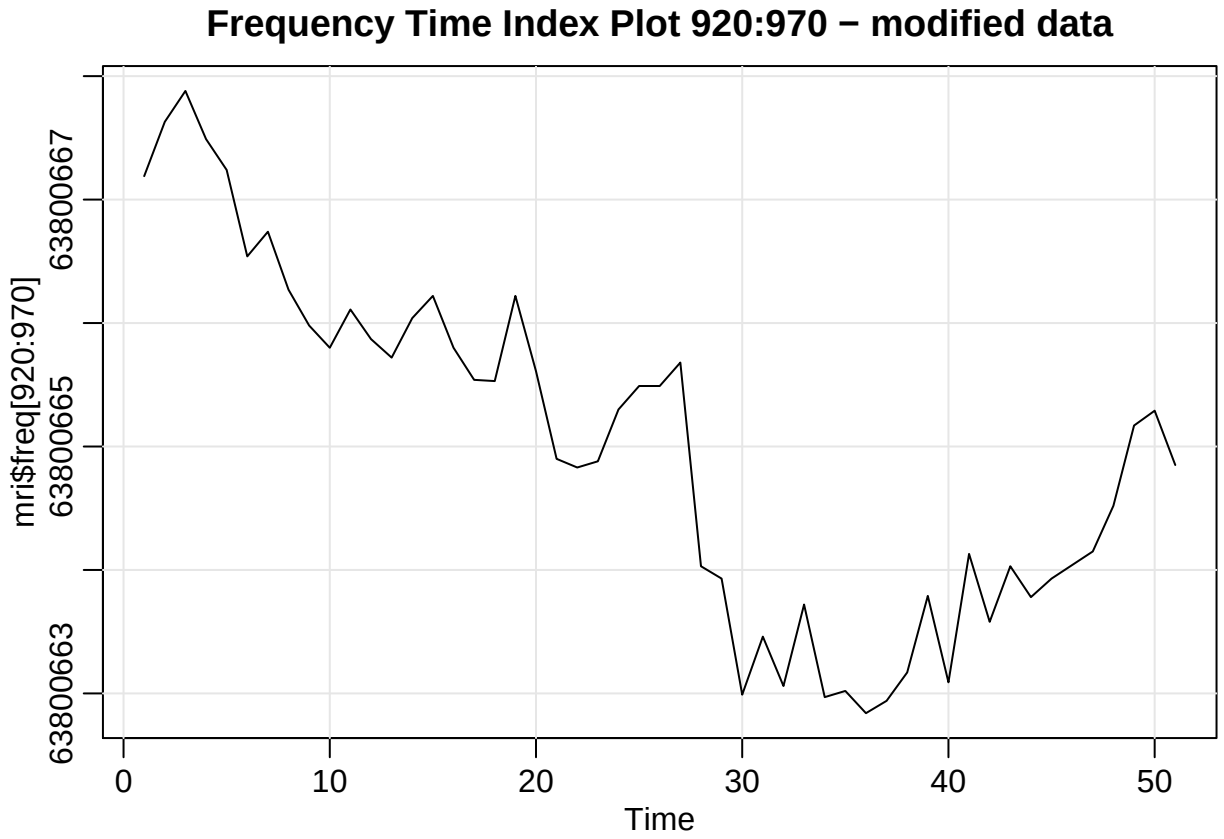
Right away - we can see that we have a data issue around index 950 that we are going to have to deal with. Let's take care of that right away.

```
astsa::tsplot(mri$freq[920:970], main="Frequency Time Index Plot 920:970")
```

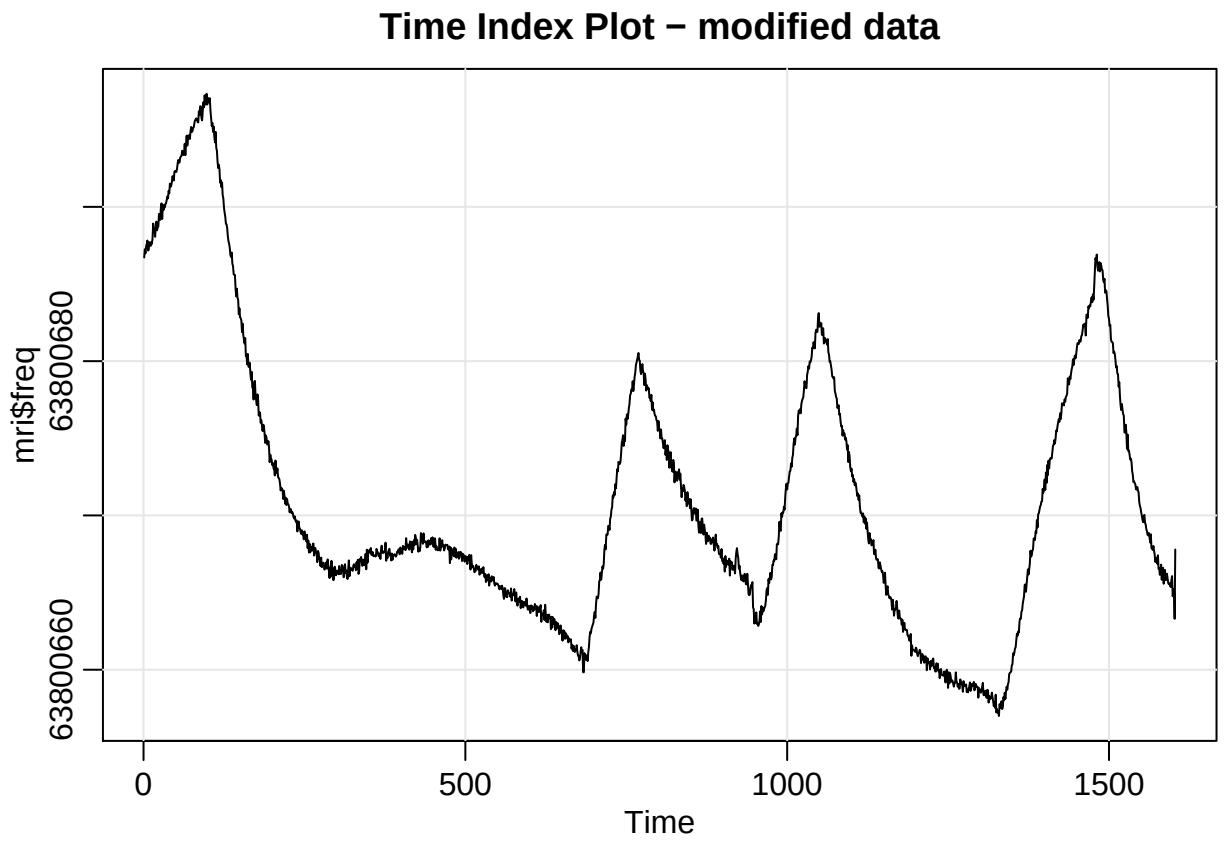


Two issues here - one massive drop in value, followed by a larger than typical value. I am going to replace the successive values with the average of the point that appeared before the abberation and the point immediately after.

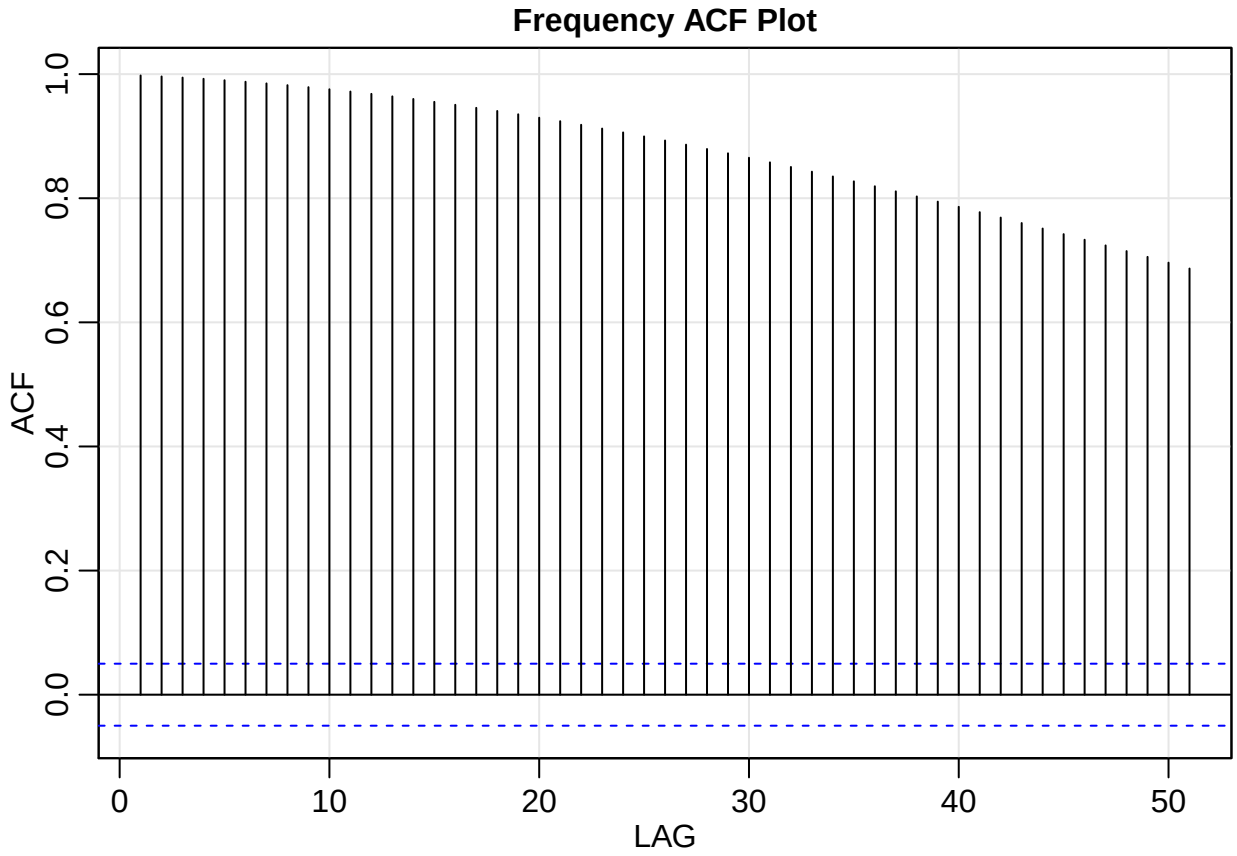
```
mri$freq[944] <- ((mri$freq[943] + mri$freq[946]) / 2)
mri$freq[945] <- ((mri$freq[943] + mri$freq[946]) / 2)
astsa::tsplot(mri$freq[920:970], main="Frequency Time Index Plot 920:970 - modified data")
```




```
astsa::tsplot(mri$freq, main="Time Index Plot - modified data")
```

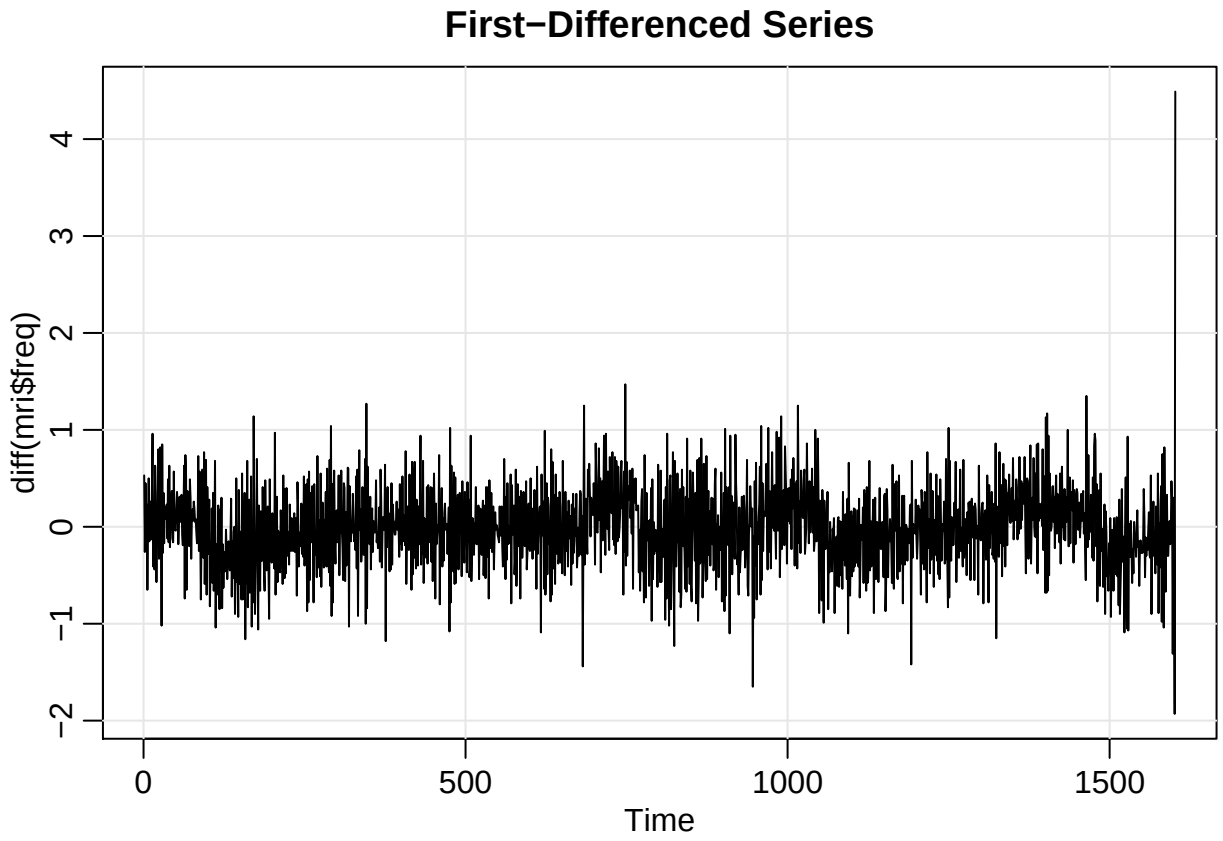


Now that we have prepped the data, we can start to assess some of its properties. Before attempting any sort of transformation, plot the ACF.



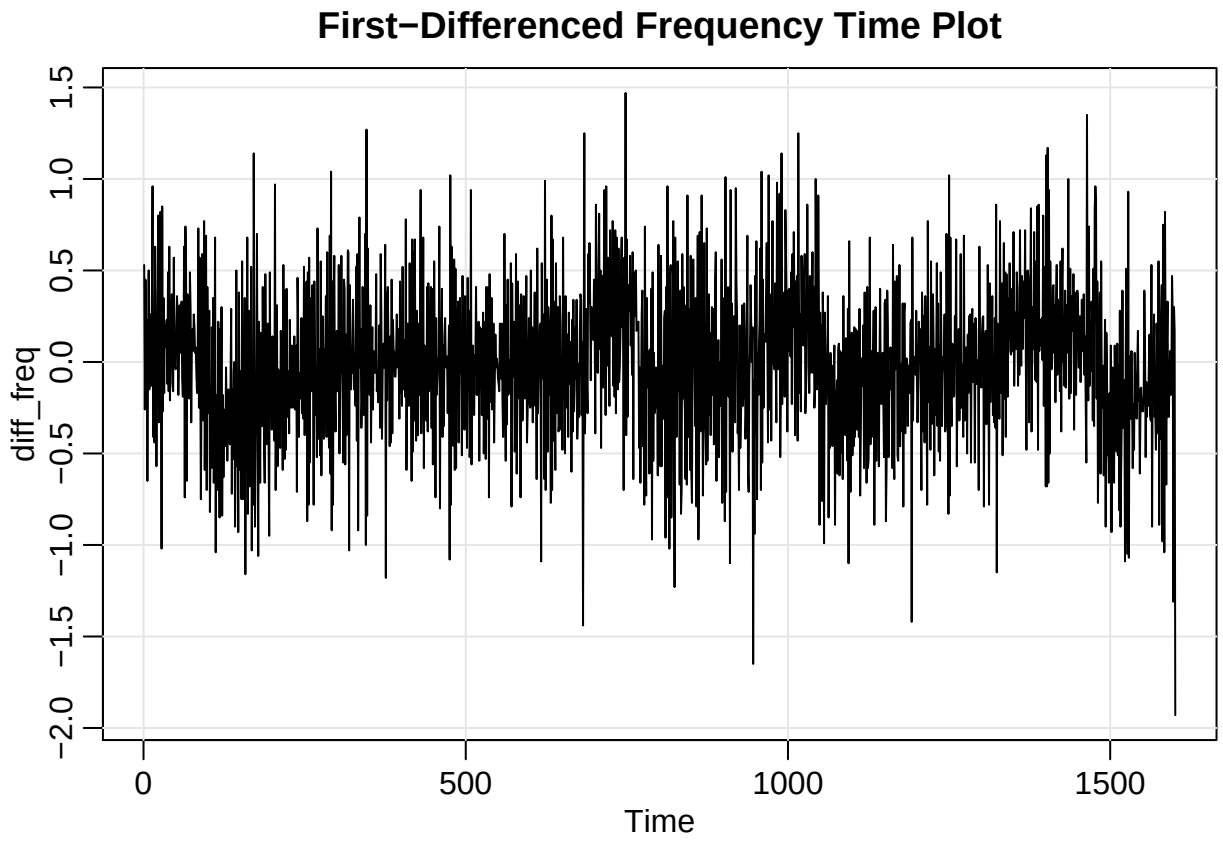
We can see from the time-plot that we have non-stationary data; from the ACF, we can see that we have an issue with autocovariance as well. Both suggest that we have a non-stationary time series. We can now employ first-differencing as a sensible technique for getting us to a stationary series.

```
astsa::tsplot(diff(mri$freq), main="First-Differenced Series")
```



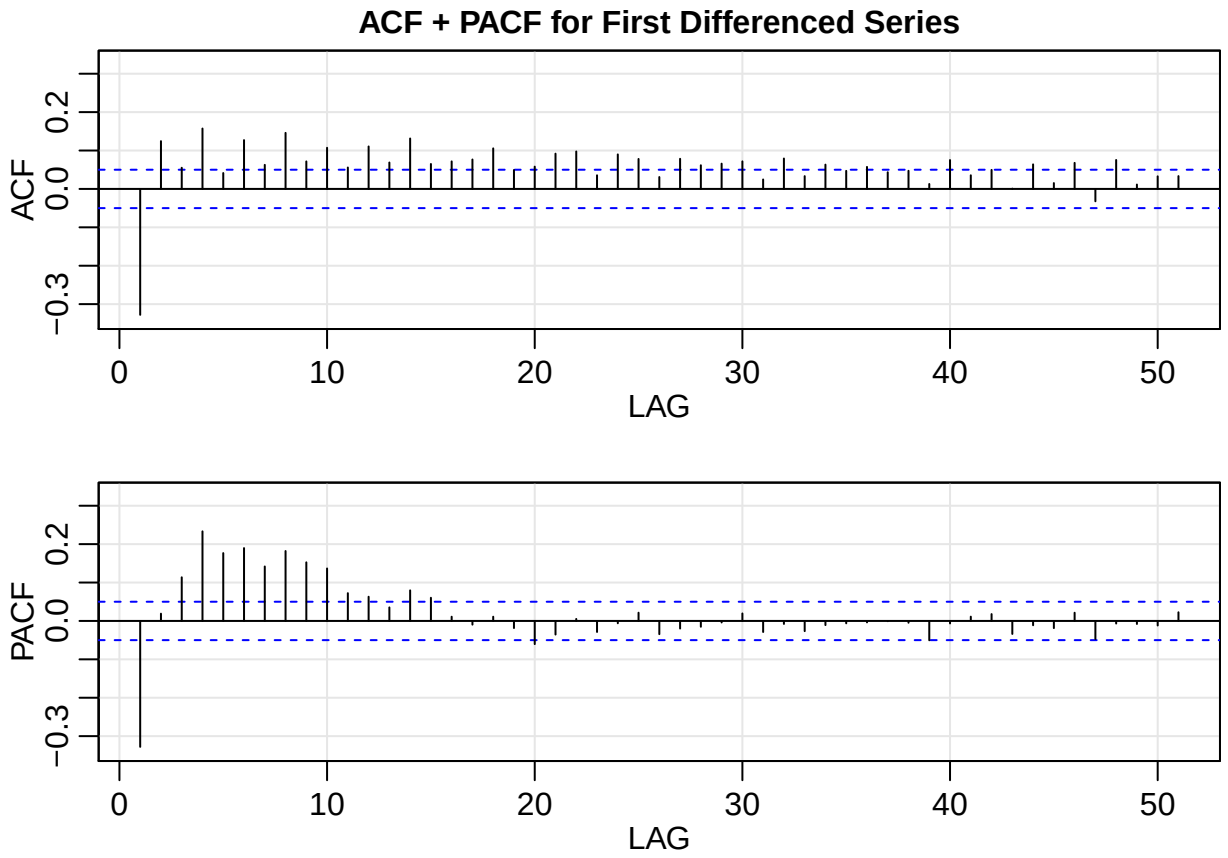
The last point is messing up the differencing, so I am going to truncate the series and leave this behind.

```
diff_freq <- diff(mri$freq)[1:(length(mri$freq)-2)]
astsa::tsplot(diff_freq, main="First-Differenced Frequency Time Plot")
```



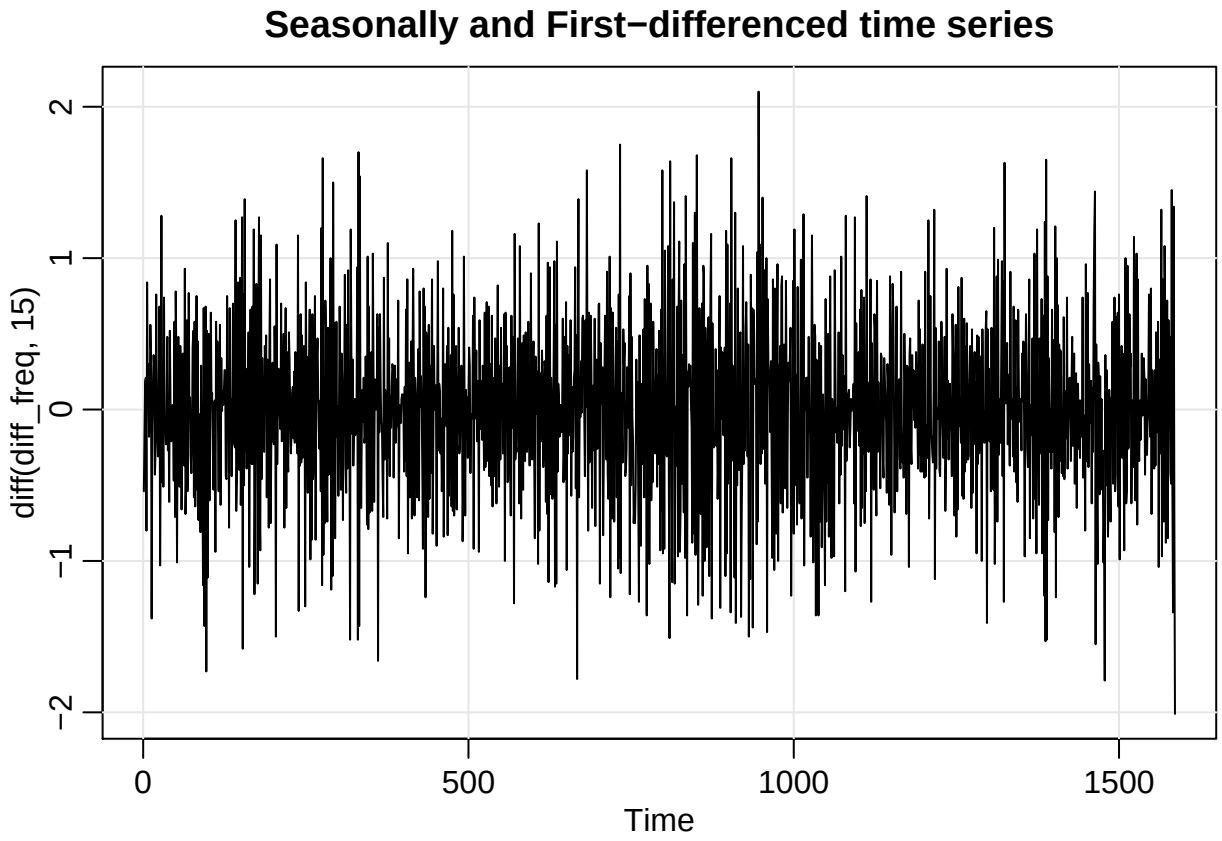
This looks a lot more stationary than what we saw before. There may be some pattern within - the ACF/PACF should help us uncover information about this.

```
astsa::acf2(diff_freq, main="ACF + PACF for First Differenced Series")
```

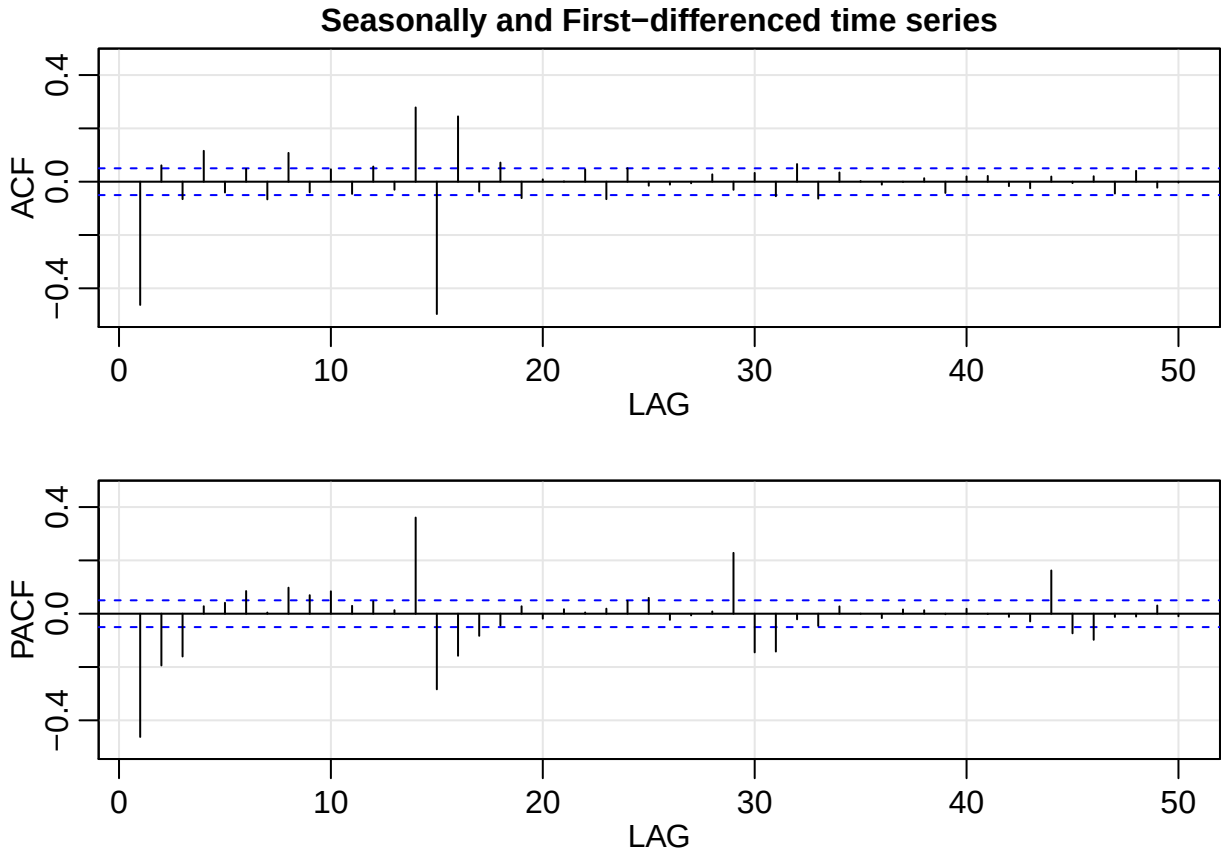


From the above, we can see that the ACF trails off, while the PACF cuts off after about lag 15. This would suggest an AR(15) model, which is the type of thing we have been advised to avoid. Let's try a "seasonal" lag of 15 seconds.

```
astsa::tsplot(diff(diff_freq, 15), main="Seasonally and First-differenced time series")
```




```
astsa::acf2(diff(diff_freq, 15), main="Seasonally and First-differenced time series")
```



Model Fitting

The ACF cuts off after around lag 1_s_ (s=15) and the PACF tails off with . This suggests an SMA model. We can easily try out a few models and see how the fits are.

```
# save the vector we care about
dd_freq <- diff(diff_freq, 15)
# lay the base to build on later - we are going to append results to this
# filled in as cbind(AR order, MA order, Seasonal Period, model AIC, model BIC)
puzzle <- cbind(0, 0, 0, 0, 0)

# seasonal order
for (k in c(15)) {
# AR order
  for (i in 0:3) {
# MA order
    for (j in 0:3) {
      # fit SARIMA model.  The optimizer will sometimes fail, so I have thrown this
      # in a try() wrapper to manage the errors and get through everything
      smod <- try(sarima(xdata=dd_freq, p=i, d=1, q=j, P=i, D=1, Q=j, S=k), silent=T)
        # if model fit successfully
        if(class(smod)=="list") {
            piece <- cbind(i, j, k, smod$AIC, smod$BIC)
                                  }
        # if model fit failed
```

```
        else {
            piece <- cbind(i, j, k, 0, 0)
             }
      # stack
      puzzle <- rbind(puzzle, piece)
    }
  }
}
```

```
## Warning in sqrt(diag(fitit$var.coef)): NaNs produced
```

```
## Warning in sqrt(diag(fitit$var.coef)): NaNs produced
```

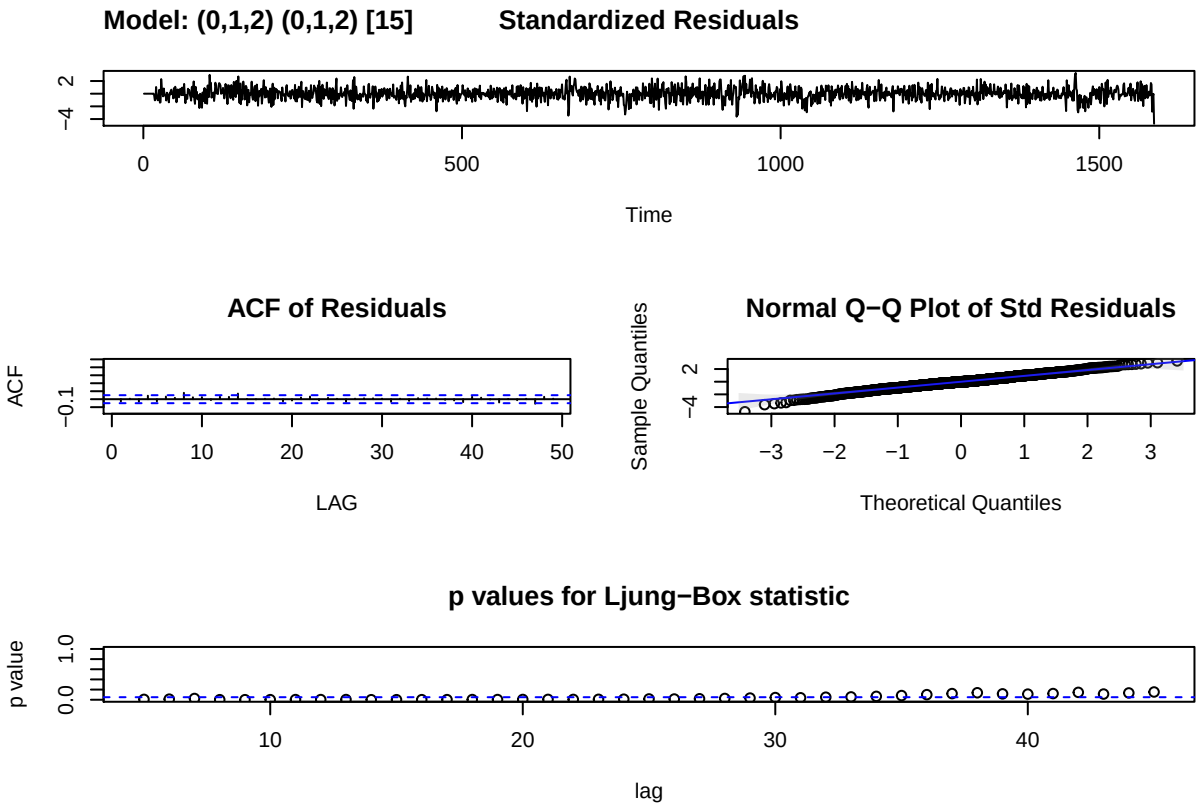
```
# nb: this will take time to run.  R is apparently bad with loops, and the try function
# takes time to work its magic
```

```
puzzle2 <- data.frame(puzzle)
names(puzzle2) <- c("i", "j", "k", "AIC", "BIC")
#
puzzle2 %>% dplyr::arrange(BIC, AIC)
```

```
##    i j  k         AIC         BIC
## 1  0 2 15 -0.94949282 -1.93595191
## 2  0 3 15 -0.95366826 -1.93335689
## 3  1 3 15 -0.95584445 -1.92876264
## 4  3 1 15 -0.72742443 -1.70034261
## 5  2 1 15 -0.64979966 -1.62948829
## 6  1 1 15 -0.50322099 -1.48968008
## 7  0 1 15 -0.01190304 -1.00513259
## 8  3 0 15  0.05778087 -0.92190777
## 9  2 0 15  0.40426452 -0.58219457
## 10 1 0 15  0.93785392 -0.05537562
## 11 0 0  0  0.00000000  0.00000000
## 12 1 2 15  0.00000000  0.00000000
## 13 2 2 15  0.00000000  0.00000000
## 14 2 3 15  0.00000000  0.00000000
## 15 3 2 15  0.00000000  0.00000000
## 16 3 3 15  0.00000000  0.00000000
## 17 0 0 15  2.15871697  1.15871697
```

From the above, it appears that SARIMA(0,1,2) x (0, 1, 2)_15 was 'best'

```
smod <- sarima(xdata=dd_freq, p=0, d=1, q=2, P=0, D=1, Q=2, S=15)
```



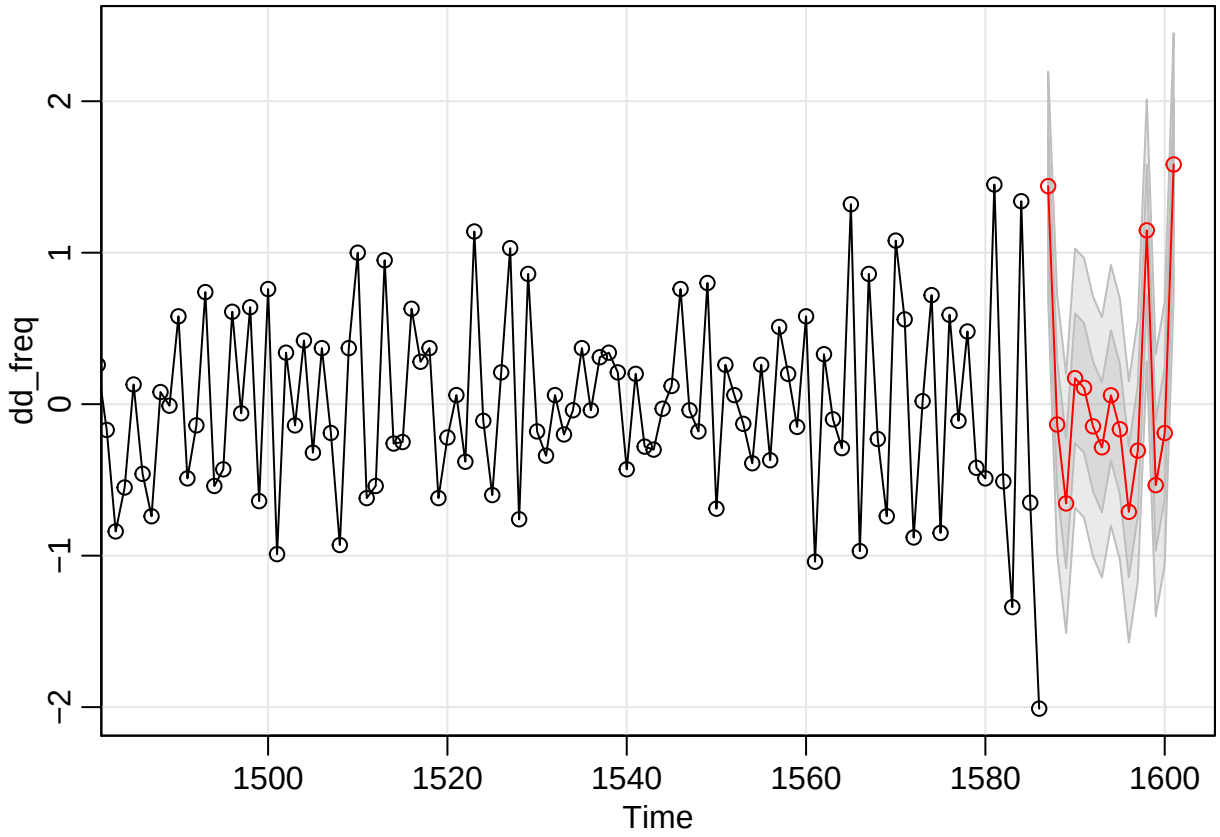

```
print(smod)
```

The diagnostics all look good, with the exception of the Ljung-Box p-values, which ideally would be > 0.05.

Forecasting

Let's take a look at what a forecast for this series would look like.

```
sarima.for(xdata=dd_freq, p=0, d=1, q=2, P=0, D=1, Q=2, S=15, n.ahead=15)
```


dd_freq
2
1
0
−1
−2
1500
1520
1540
1560
1580
1600
Time